CLAIMS

1. A parallel merge/sort processing device, comprising:

parallel processors for merging/sorting that includes a plurality of processors for executing merge/sort processing for two input data strings;

parallel processors for dividing data string pairs for sub-dividing two sorted data string pairs to be the input of the merge/sort processing into a specified number of sets of partial data string pairs; and

a processor for management for controlling the entire system, characterized in that said processor for management assigns a plurality of partial data string pairs received from said parallel processor group for division to said parallel processors for merging/sorting respectively to execute merge processing in parallel.

2. The parallel merge/sort processing device according to Claim 1, characterized in that said parallel processors for merging/sorting to which a pair of partial data strings is assigned execute the merge processing independently and output the result to a pre-instructed output area, and the whole of these output areas become the final merge result or is output as an intermediate merge result so as to be used for the parallel merging in the next step.

3. The parallel merge/sort processing device according to Claim 2, characterized in that two processors are assigned to said partial data string pairs respectively, the first processor executes merge processing in descending order (merging said partial data string pair from the edge at which the key value is greater to the side at which the key value is smaller, and

writing the output sequentially from the edge at which a key value is greater to the side at which the key value is smaller in said output area), and the second processor executes the merge processing in ascending order (merging said partial data string pair from the edge at which a key value is smaller to the side at which a key value is greater, and writing the output sequentially from the edge at which the key value is smaller to the side the key value is greater in said output area).

4. The parallel merge/sort processing device according to Claim 1, characterized in that the following items are satisfied as said division conditions:

the sorted data string D comprised of n number of data is expressed as (D, n) and a pair of two data strings is expressed as $\{(D_1, n), (D_2, n)\}$;

when the sorted data string pair $\{(D_1, n), (D_2, n)\}$ is divided into two partial data string pairs $\{(D_{11}, n_{11}), (D_{21}, n_{21})\}$ and $\{(D_{12}, n_{12}), (D_{22}, n_{22})\}$, (smaller one of the key values of the last part of the data in the partial data strings D_{11} and D_{21}) $\geq$ (greater one of the key values of the first part of the data in the partial data strings D_{12} and D_{22}) is established, and also

$n_{11} + n_{21} = 2x$, $n_{12} + n_{22} = 2(n-x)$

is established, where x is half the value of the number of data counted from the first part of D_1 and D_2 in the area pair.

5. A method for performing parallel merge/sort processing using a parallel processor, where one unsorted data string, or two sorted partial data strings, or one of three or more sorted partial data strings and attribute information thereof (e.g. type, location information of data, and number of data) are received as the input, comprising:

a first step of dividing a data string and acquiring a plurality of unsorted partial data strings, if the sorting target is an unsorted data string;

a second step of assigning a processor to said plurality of partial data strings respectively;

a third step of sorting each of said plurality of partial data strings independently by the assigned processor based on an arbitrary algorithm, such as a quick sort method, to acquire sorted partial data strings;

a fourth step of creating an input data string pair for merging/sorting using arbitrary two sorted partial data strings which were acquired in the third step or the seventh step or was input as initial data, and dividing the pair into required number of sets of partial data string pairs respectively under a predetermined division condition;

a fifth step of editing job information for parallel merging/sorting from the plurality of divided partial data string pairs;

a sixth step of assigning a processor group for merging/sorting to the acquired job for parallel merging/sorting;

a seventh step of performing merge processing by the assigned processor; and

a step of repeating said fourth step to said seventh step using the merge-processed sorted partial data strings as said sorted partial data strings, characterized in that the above steps are applied to the case when one unsorted data string is provided, and the first to the third steps are unnecessary if two or more sorted data strings are provided, and the repeat of said fourth to seventh steps ends in the stage when the merge-processed sorted partial data strings are merged into one data string. If two sorted data strings are provided,

the fourth to seventh steps are executed only once, and a repeat is not required.

6. The parallel merge/sort processing method according to Claim 5, characterized in that two processors are assigned to said partial data string pair in said sixth step, and the job control information is edited in the fifth step so that the first processor performs merge processing in descending order from the side of which a key value is greater in said partial data string and the second processor performs merge processing in ascending order from the edge of which a key value is smaller in the same partial data string simultaneously in said seventh step.

7. The parallel merge/sort processing method according to Claim 5, characterized in that the following items are satisfied as the division condition in said fourth step, wherein

when a data string pair (D_1, n) and (D_2, n) are divided into two partial data string pairs $\{(D_{11}, n_{11}), (D_{21}, n_{21})\}$ and $\{(D_{12}, n_{12}), (D_{22}, n_{22})\}$, (smaller one of the key values of the last part of the data in the partial data strings D_{11} and D_{21}) $\geq$ (greater one of the key values of the first part of the data in the partial data strings D_{12} and D_{22}) is established, and also

$n_{11} + n_{21} = 2x$, $n_{12} + n_{22} = 2(n - x)$

is established, where x is a half value of the number of data of the partial data string pair $\{(D_{11}, n_{11}), (D_{21}, n_{21})\}$, and is also the number of data of D_{11} and D_{21} when $n_{11} = n_{21}$.

8. The parallel merge/sort processing method according to Claim 5, characterized in that said fourth step has the following functions:

(1) An operation to divide a sorted data string pair $\{(D_1, n), (D_2, n)\}$ into k sets of segment pairs, which is equivalent to performing (k-1) sets of two-division operations in which the total of the number of data counted from the first part of D_1 and D_2 becomes 2x with changing the value of x, while considering the magnitude of the key values of both data strings. In this case, the sub-division problem of the sorted data string pair $\{(D_1, n), (D_2, n)\}$ to the k sets of segment pairs is replaced with the above-mentioned two-division problem of the data string that satisfies Claim 7.

(2) Specifying the data position in the data string by an index value. This value sequentially increments as 1, 2, . . . with the index value of the first data in the data string D_1 or D_2 as 0. x in Claim 7 indicates the number of data, but if the value of x itself is regarded as an index value, then [x] indicates the (x+1)th data counted as 1, 2, 3, . . . from the first part of the data string. If $n_{11} = n_{21}$ in Claim 7, then $n_{11} = n_{21} = x$, which is a formula indicating the number of data, can be interpreted that the position of the xth data counted from the first part, that is data with the index value x-1, is at the division boundary of D_1 and D_2.

(3) An area dividing function, comprising:

a step of setting said x as an initial value of the boundary index value for the index variables i and j for specifying individual data in said data strings D_1 and D_2 (e.g. when the data string pair, with 10,000 data in each data string, is divided into 10 segments of data string pairs, with 1000 data in each data string, x = 1000 is set as the number of data, and 1000, . . . 9000 is set for the initial index of division positions);

a comparison step of comparing a key value of data indicated by the index variable i of the data string D_1 and a

key value of data indicated by the index variable j of the data string D_2;

a step of adding 1 to an index variable of the data with a greater key value, subtracting 1 from an index variable of the data with a smaller key value, then branching processing to said comparison step, if the key value of the data indicated by the index variable i of D_1 and the key value of the data indicated by the index variable j of D_2 are not the same in the initial comparison;

a step of adding 1 to the index variable of data with a greater key value, and subtracting 1 from the index variable of data with a smaller key value, then branching processing to said comparison step, if the magnitude relationship of the key value of data indicated by the index variable i of D_1 and the key value of data indicated by the index variable j of D_2 is unchanged in the second or later comparison;

a step of regarding the data indicated by the index variable i and the data indicated by the index variable j as a division boundary respectively, if the key value of the data indicated by the index variable i of D_1 and the key value of the data indicated by the index variable j of D_2 are the same in the initial comparison; and

a step of comparing the greater one of the key value of D_1 and the key value of D_2 in the previous comparison operation with the greater one of the key value of D_1 and the key value of D_2 in the current comparison operation, and regarding the data with a smaller key value as the division boundary and regarding the data initially compared with this data as the other boundary, if the magnitude relationship between the key value of the data indicated by the index variable i of D_1 and the key value of the data indicated by the index variable j of D_2 is inverted from the previous magnitude relationship (the case when the key

values are the same is also regarded as an inversion in the second or later comparison).

This parallel merge/sort processing method is used for determining the position of the first part of an insignificant data string pair after two-division is performed.

9. The parallel merge/sort processing method according to Claim 5, characterized in that said fourth step has the following functions:

(1) An operation to divide a sorted data string pair $\{(D_1, n), (D_2, n)\}$ into k sets of segment pairs, which is equivalent to performing (k-1) sets of two-division operations in which the total of the number of data counted from the first part of D_1 and D_2 becomes 2x with changing the value of x, while considering the magnitude of the key values of both data strings. In this case, the sub-division problem of the sorted data string pair $\{(D_1, n), (D_2, n)\}$ to the k sets of segment pairs is replaced with the above-mentioned two-division problem of the data string that satisfies the above-mentioned conditions in Claim 7.

(2) Specifying the data position of the data string by an index value. This value sequentially increments as 1, 2, . . . with the index value of the first data in D_1 or D_2 as 0. x in Claim 7 indicates the number of data, but if the value of x itself is regarded as an index value, then [x] indicates the (x+1)th data counted as 1, 2, 3. . . from the first part in the data strings. If $n_{11} = n_{21}$ in Claim 7, then $n_{11} = n_{21} = x$ indicates that the division boundary of D_1 and D_2 exists at the position of the xth data counted from the first part, that is at the data position of the index value x-1.

(3) An area division function, comprising:

a step of setting said x-1 as an initial value of the boundary index value for the index variables i and j for specifying individual data in said data strings D_1 and D_2 (e.g. when the data string pair, with 10,000 data in each data string, is divided into 10 segments of data string pairs, with 1000 data in each data string, 1000 is set for x as the number of data, and 999, 1999 . . . 8999 is set for the initial index of the division position);

a comparison step of comparing a key value of data indicated by the index variable i of the data string D_1 and a key value of data indicated by the index variable j of the data string D_2;

a step of adding 1 to an index variable of the data with a greater key value, subtracting 1 from an index variable of the data with a smaller key value, then branching processing to said comparison step, when the key value of the data indicated by the index variable i of D_1 and the key value of the data indicated by the index variable j of D_2 are not the same in the initial comparison;

a step of adding 1 to an index variable of the data with a greater key value, subtracting 1 from an index variable of the data with a smaller key value, then branching processing to said comparison step, if the magnitude relationship of the key value of the data indicated by the index variable i of D_1 and the key value of the data indicated by the index variable j of D_2 is unchanged in the second or later comparison;

a step of regarding the data indicated by the index variable i and the data indicated by the index variable j as a division boundary respectively, when the key value of the data indicated by the index variable i of D_1 and the key value of the data indicated by the index variable j of D_2 are the same in the initial comparison; and

a step of comparing the smaller one of the key value of D_1 and the key value of D_2 in the previous comparison operation with the smaller one of the key value of D_1 and the key value of D_2 in the current comparison operation, and regarding the data with a greater key value as the division boundary, and regarding the data initially compared with this data as the other boundary, if the magnitude relationship between the key value of the data indicated by the index variable i of D_1 and the key value of the data indicated by the index variable j of D_2 is inverted from the previous magnitude relationship (the case when the key values are the same is also regarded as an inversion in the second or later comparison).

This parallel merge/sort processing method is used for determining the last part of a significant data string pair after two-division is performed.

10. A parallel merge/sort processing method for sub-dividing two sets of sorted data string pairs into a plurality of sorted data string pairs so as to enable sort by a merge/sort operation at an arbitrary parallelism, comprising:

a step of providing a pair of first sorted data string and a second sorted data string;

a division step of sub-dividing the pair of said first data string and said second data string into a plurality of sorted partial data string pairs according to a required parallelism;

and a step of merging the sub-divided sorted partial data string pairs in parallel, characterized in that data strings without any inconsistency in the entire key arrangement can be output regardless the logorithm of the sub-divided sorted partial data string.

11. A program for performing parallel merge/sort processing using a parallel processor which includes a plurality of processors, where one unsorted data string, or two sorted partial data strings, or one of three or more sorted partial data strings and attribute information thereof (e.g. type, location information of data and number of data) are received as an input, comprising:

a first step of dividing a data string and acquiring a plurality of unsorted partial data strings when the sorting target is an unsorted data string;

a second step of assigning a processor to said plurality of partial data strings respectively;

a third step of sorting each of said plurality of partial data strings independently by the assigned processor based on an arbitrary algorithm, such as a quick-sort method;

a fourth step of creating an input data string pair for merging/sorting using arbitrary two sorted partial data strings which were acquired in the third step or the seventh step or was input as initial data, and dividing the pair into the required number of sub-divided partial data string pairs respectively under a predetermined division condition;

a fifth step of editing job information for merging/sorting the divided partial data string pairs;

a sixth step of assigning a processor group to the acquired merge/sort job;

a seventh step of performing merge processing in parallel by the assigned processors; and

a step of repeating said fourth step to seventh step using the merge-processed data strings as said partial data strings, characterized in that the above is applied when one unsorted data string is provided, and the first step to the third step are unnecessary if two or more sorted data strings

are provided, and the repeat of said fourth step to said seventh step ends when the merge-processed sorted partial data strings become one data string. The fourth step to the seventh step are executed once, and a repeat is not required if two sorted data strings are provided.